WES/GPIB-MI Interface User's Guide

Table of Contents, Cont'd

Table of Contents, Cont'd
List of Figures

Table of Contents, Cont'd
List of Tables

Section 1. Introduction

1-1. Overview

This document describes the operation of the WES/GPIB/MHI Interface, hereafter referred to as the MHIGPIB Interface.

The Mitsubishi™ General Purpose Interface Bus (GPIB) Interface provides a link between the WDPF® Westnet II™ Data Highway and the Mitsubishi Heavy Industries (MHI) Digital Intelligent Automation System (DIASYS). The interface software will run on any WEStation with a National Instrument Corporation SB-GPIB/TNT™ SBus card installed.

The MHIGPIB Interface allows data to flow in both directions (see Figure 1-1). The WEStation on the Westnet II data highway communicates with the MHI DIASYS via IEEE 488 cable(s). Elements of the standard IEEE 488.2 communication protocol are used. Exchange of the following types of data is possible:

- Analog point values
- Digital point values

Figure 1-1 illustrates how data flows between the WDPF and the MHI DIASYS systems.

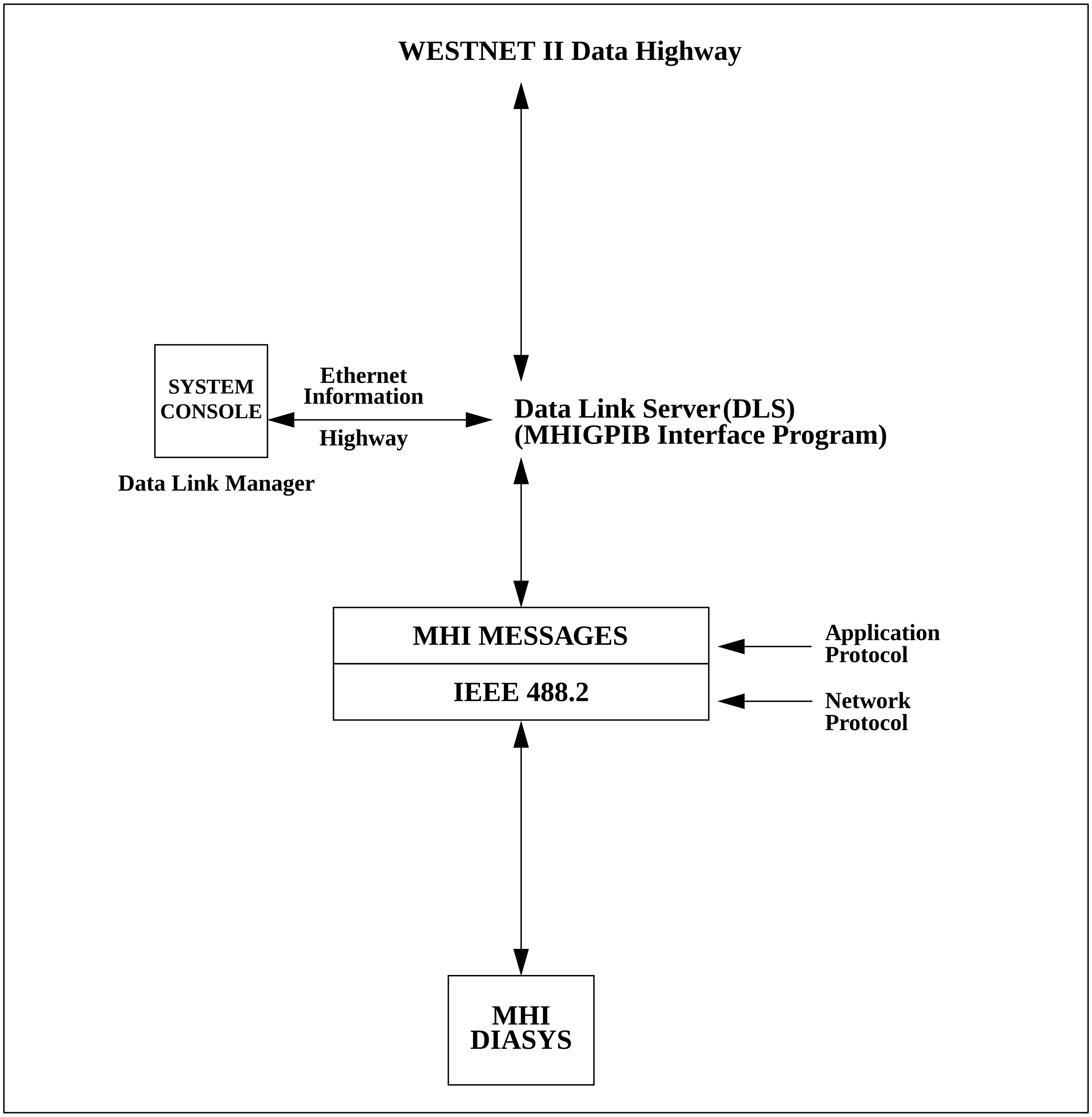


Figure 1-1. Data Flow Diagram

1-2. Contents of this Document

This document is organized into the following sections:

Section 1. Introduction provides an overview of the MHIGPIB Interface.

Section 2. Hardware describes the hardware required for the MHIGPIB Interface, and how the interface is configured.

Section 3. Software Installation discusses the required interface software and how it is installed.

Section 4. Software Configuration discusses the configuration files used by the MHIGPIB Interface.

Section 5. Communication Protocol discusses the communication protocol and message structure for the MHIGPIB Interface.

Section 6. Interface Operation discusses the startup, operation, diagnostic capabilities, and shutdown of the software.

Appendix A. Installing the NI-488.2M Driver provides detailed instructions for installing the NI-488.2M software driver.

1-3. Reference Documents

Table 1-1 lists additional reference documentation that may be helpful in understanding the MHIGPIB Interface.

Table 1-1. Reference Documents

Document Number	Title	Description
M0-0003	Self-Test Diagnostics	Describes the self-test diagnostics incorporated in the WDPF system. Includes system-level and drop-level diagnostics, fault codes, and instructions for accessing diagrams on the Operator Station.
U0-0131	Record Types User's Guide	Describes point, system and algorithm record types and the purpose and use of the records. Used as a reference for drop application programming.
U0-8100	Operator WEStation User's Guide	Describes the use of Operator WEStation windows for viewing and controlling the current status of plant process control systems.
U0-8200	Engineering WEStation User's Guide	Describes the functions of the Engineering WEStation drop for the development and maintenance of application and system software.
U0-8700	WEStation Data Link Server User's Guide	Describes the configuration and operation of the WEStation Data Link Server.
ANSI/IEEE 488.1-1987	Digital Interface for Programmable Instrumentation	Describes electrical and mechanical specifications for GPIB interfaces.
ANSI/IEEE 488.2-1987	Codes, Formats, Protocols, and Common Commands	Describes common commands, status reporting, data formats, message exchange protocol, protocols, and control sequences for GPIB interfaces.

Section 2. Hardware

2-1. Section Overview

This section describes the hardware required for the MHIGPIB Interface. It is divided into the following sections:

- WEStation on the Westnet II Data Highway (Section 2-2)
- Communication Hardware (Section 2-3)
- Hardware Configuration (Section 2-4)

2-2. WEStation on the Westnet II Data Highway

The MHIGPIB Interface software will run on any WEStation with an available GPIB parallel device driver (supplied through National Instruments Corp.).

The MHIGPIB Interface requires a text editor to create the configuration files that define the point groups. Any WEStation equipped with a text editor can be used to perform this task (for example, an Engineering WEStation, also known as a Software Development Station).

Note

An Engineering WEStation provides direct access to the applicable operating system, other drops, and the Westnet II Data highway. Therefore, it is assumed that the user has a thorough understanding of the functionality of the WDPF systems and the operating system commands before using the Engineering WEStation functions (as described in "Engineering WEStation User's Guide" (U0-8200)).

For further information regarding the hardware configuration, refer to IEEE 488 Hardware documentation (ANSI/IEEE 488.1-1987).

2-2.1. Data Link Server

Data Link Server (DLS) refers to a WEStation that is running data link subsystems and is connected to the Ethernet Information Highway and the WDPF Westnet II Data Highway. Its purpose is to accept and maintain a communication link from the client (remote computer device) and the Westnet II Data Highway. The client uses this link to access information from the WDPF process control system. Process point information can be read from or originated to the Westnet II Data Highway.

For additional information regarding the WEStation Data Link Server and its functions, refer to the "WEStation Data Link Server User's Guide" (U0-8700).

2-3. Communication Hardware

In addition to a WEStation, the MHIGPIB Interface requires the following hardware:

- GPIB-SPRC-B Package which includes:
  - SB-GPIB/TNT SBus Card
  - GPIB Connection Extender
  - Software Device Driver (National Instrument's NI-488.2M™ for Solaris™ 2 (see Section 3))
- IEEE 488 Cable(s)

The GPIB-SPRC-B Package can be purchased from National Instruments (Part Number 776789-01).

The IEEE 488 cable should be single shielded, 4 meters in length. There should be one IEEE 488 cable per GPIB link. The IEEE 488 cable can also be purchased from National Instruments (Part Number 763001-xx, where xx indicates the length in meters, 01 to 04).

Note

As of publication, it was not possible to purchase single-shielded cable. If single-shielded cable is absolutely required, then the metal shield at the connector that attaches to the MHI DIASYS must be cut off at installation.

2-4. Hardware Configuration

The SB-GPIB/TNT card is inserted into the SBus master slot on the back of the WEStation. The GPIB connection extender is attached to the installed GPIB board connector. One end of the IEEE 488 cable connects to the GPIB connection extender, and the other end connects to the MHI DIASYS equipment.

For more information about GPIB-SPRC-B card installation and configuration, refer to National Instrument’s “Getting Started with Your SB-GPIB/TNT” and “NI-488.2M Software for Solaris 2.”

2-4.1. Redundancy

Redundant configuration is possible for the MHIGPIB Interface. It is currently configured for two links, but up to four redundant links are possible. Redundancy is accomplished by installing additional GPIB (IEEE 488) links and starting up additional **dl_mhigpib** link drivers with different subsystem numbers. Each link has its own error checking and correcting capability. If all links go down for any reason, then all points being transferred from the MHI DIASYS to the Westnet II Data Highway will have their status changed to Bad. Otherwise, the status of these points reflects the status sent over from the MHI DIASYS.

Figure 2-1 illustrates a redundant system configuration.

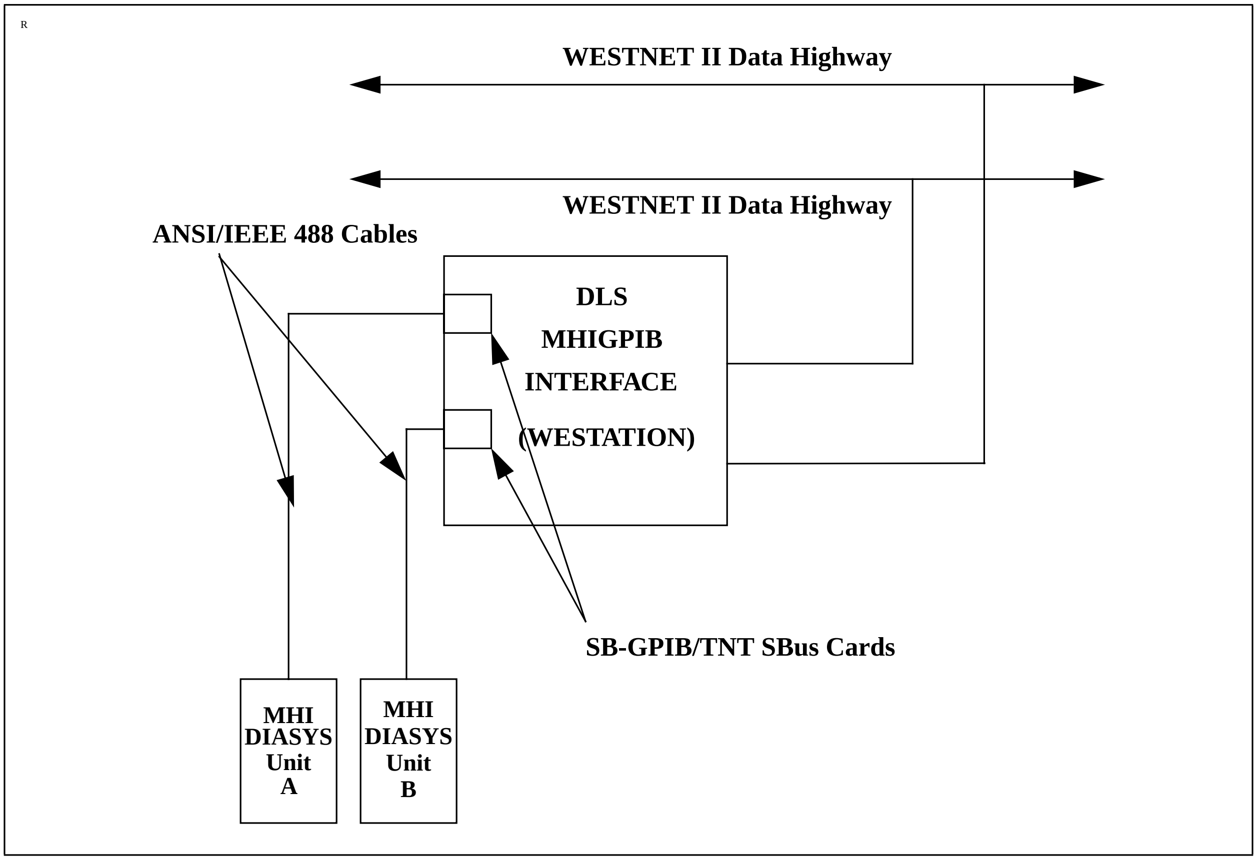


Figure 2-1. Redundant Configuration

Section 3. Software Installation

3-1. Section Overview

This section discusses the software requirements and installation for the MHIGPIB Interface. It is divided into the following sections:

- Software Requirements (Section 3-2)
- Software Installation (Section 3-3)

3-2. Software Requirements

The software required for the MHIGPIB Interface is the following:

- National Instrument GPIB Software Driver (NI-488.2M, Version 1.3b or later) (provided with National Instrument's GPIB-SPRC-B package discussed in Section 2)
- Westinghouse MHI DLS software package (WESTdlgp)
  - — Standard Westinghouse configuration file (CONFIG.DL)
  - — Group point file with two sets of points (MHIGPIB.GROUP)

3-3. Installation Procedure

This section discusses installation of the MHIGPIB Interface software.

3-3.1. NI-488.2M Software Driver

The NI-488.2M driver (Version 1.3b or later) requires a Solaris 2.x operating system. For this version of the MHIGPIB Interface, use the installation procedure provided in Appendix A. For additional information or future revisions, follow National Instrument Corporation’s installation material.

3-3.2. WESTdlgp Software

The MHIGPIB Interface software is contained in the WESTdlgp package. The following sequence of commands is used to install the software.

1. Copy the contents of the tape to a temporary directory:

```
cd /tmp

tar -xvf /dev/<device name>
```

2. Add the WESTdlgp package as follows:

```
pkgadd -d /tmp
```

   Root privileges (see system administrator) are needed to perform the **pkgadd** command. The executables and examples of the configuration file will be installed. These example configuration files can be used as templates to customize the parameters.

Section 4. Software Configuration

4-1. Section Overview

This section discusses the software setup and configuration for the MHIGPIB Interface. It is divided into the following sections:

- System Setup (Section 4-2)
- Configuration Files (Section 4-3)

4-2. System Setup

This section describes the MHIGPIB Interface directory structure.

4-2.1. Directory Structure for the Interface Software

The directory structure for the MHIGPIB Interface is as follows:

The subdirectory **dl** is located under the **$(WDPF_HOME)** directory. The environment variable, **$WDPF_HOME**, is defined by the software package. It is usually set to **/usr/wdpf**.

The directory **$(WDPF_HOME)/dl** contains the following three subdirectories:

- **init**
- **config**
- **bin**

The **$(WDPF_HOME)/dl/init** directory contains the startup and shutdown script files as follows:

- **startup.dl**
- **start_datalinks**
- **shutdown.dl**

The **$(WDPF_HOME)/dl/config** directory contains the configuration and point group files, as follows

- **CONFIG.DL** (configuration file) - see Section 4-3.1
- **MHIGPIB.GROUP** (point group file) - see Section 4-3.3

The **$(WDPF_HOME)/dl/bin** directory contains the binary executable, as follows:

- **dl_mhigpib -** see Section 4-2.2

The installation procedure will install the files in the correct directories.

4-2.2. Executable File (dl_mhigpib)

The binary executable, **dl_mhigpib**, resides in directory $(WDPF_HOME) /dl/bin.

Multiple instances of the executable file can run simultaneously on the same machine. This allows the DLS to communicate with more than one MHI DIASYS unit at a time. A group of parameters in the CONFIG.DL file is matched with its corresponding executable file through the use of a command line argument. For example, the executable must be entered as **dl_mhigpib -l 2** to use the parameters that begin with DL.MGPIB.2.

4-3. Configuration Files

The MHIGPIB Interface uses two separate configuration files:

- CONFIG.DL
- MHIGPIB.GROUP

4-3.1. CONFIG.DL File

The Data Link Configuration File (CONFIG.DL) is used by all data links running on a WEStation. It is read when the link starts up. The CONFIG.DL file defines the parameters of the data link and the diagnostic points that are used to report the current status of the link. It defines attributes such as point group configuration file name, the last transmit time point, message time-out values, and syslog priority. The CONFIG.DL file is used to get the name of each link and its associated link status point. The CONFIG.DL file is further described in “WEStation Data Link Server User’s Guide” (U0-8700).

If multiple copies of the MHIGPIB Interface are run, then there should be multiple entries. The entries should be labeled as: DL.MGPIB.1, DL.MGPIB.2, DL.MGPIB.3, and so on.

If different point group files are to be used, then the **DL.MGPIB.x.PointGroupDefFile** entry should be unique for each of the different point group files.

Figure 4-1 provides an example of a CONFIG.DL file for the MHIGPIB Interface. The default value is shown for each parameter.

```
DL.MPGIB.1.SyslogPriority     : 3
DL.MGPIB.1.PointGroupDefFile  : MHIGPIB.GROUP
DL.MGPIB.1.LinkStatusPoint    : L1LASTST
DL.MGPIB.1.LastCommPoint      : L1LASTCOMM
DL.MGPIB.1.DigitalAlarmPoint0 : L1DIGALARM
DL.MGPIB.1.DigitalAlarmPoint1 : L3DIGALARM
DL.MGPIB.1.TestMsgFile        : stderr
DL.MGPIB.1.RestartProcedure   : $WDPF_HOME/dl/bin/dl_mhigpib -l 1
DL.MGPIB.1.LinkName           : MHI dls 1 gateway
DL.MGPIB.1.LocalHost          : drop130
DL.MGPIB.1.DeviceName         : /dev/dev1
DL.MGPIB.1.Timeout            : 500
DL.MGPIB.1.Delay              : 0
DL.MGPIB.1.IOTMO              : 9
DL.MGPIB.1.NEOI               : 1
DL.MGPIB.1.READDR             : 1
DL.MGPIB.1.EOS                : 0
DL.MGPIB.1.EOS_EOI            : 0
DL.MGPIB.1.EOS_SZ             : 0
DL.MGPIB.1.EOS_CHR            : 0
DL.MGPIB.1.BENDRD             : 0
DL.MGPIB.1.BENDWR             : 0
DL.MGPIB.1.UNADDR             : 1
DL.MGPIB.1.BYTSPS             : 1
DL.MGPIB.1.Cable_len          : 0
DL.MGPIB.1.Xfer_Protocol      : 0

DL.MGPIB.2.SyslogPriority     : 3
DL.MGPIB.2.PointGroupDefFile  : MHIGPIB.GROUP
DL.MGPIB.2.LinkStatusPoint    : L2LASTST
DL.MGPIB.2.LastCommPoint      : L2LASTCOMM
DL.MGPIB.2.DigitalAlarmPoint0 : L2DIGALARM
Dl.MGPIB.2.DigitalAlarmPoint1 : L4DIGALARM
DL.MGPIB.2.TestMsgFile        : stderr
DL.MGPIB.2.RestartProcedure   : $WDPF_HOME/dl/bin/dl_mhigpib -l 2
DL.MGPIB.2.LinkName           : MHI dls2 gateway
DL.MGPIB.2.LocalHost          : drop130
DL.MGPIB.2.DeviceName         : /dev/dev2
DL.MGPIB.1.Timeout            : 500
Dl.MGPIB.2.Delay              : 0
DL.MGPIB.2.IOTMO              : 9
DL.MGPIB.2.NEOI               : 1
DL.MGPIB.2.READDR             : 1
DL.MGPIB.2.EOS                : 0
DL.MGPIB.2.EOS_EOI            : 0
DL.MGPIB.2.EOS_SZ             : 0
DL.MGPIB.2.EOS_CHR            : 0
DL.MGPIB.2.BENDRD             : 0
DL.MGPIB.2.BENDWR             : 0
DL.MGPIB.2.UNADDR             : 1
DL.MGPIB.2.BYTSPS             : 1
DL.MGPIB.2.Cable_len          : 0
DL.MGPIB.2.Xfer_Protocol      : 0
```

Figure 4-1. Example of CONFIG.DL File

Table 4-1 defines the parameters in the MHIGPIB Interface. The format is DL.MGPIB.x.LinkName, where x = the link number.

Table 4-1. MHIGPIB Interface Configuration File Parameters

Parameter	Description	Valid Option	Default
DL.MGPIB.x. SyslogPriority	Specifies priority of the error messages to be logged	3 through 7 (See Table 6-1 for details)	3
DL.MGPIB.x. PointGroupDefFile	Point group file name	Any valid file name in dl/config directory	MHIGPIB.GROUP
DL.MGPIB.x. LinkStatusPoint	Status point name	Name of originated PB point	L1LASTST
DL.MGPIB.x. LastCommPoint	Indicates date and time of the last successful communication message	Name of originated analog point	L1LASTCOMM
DL.MGPIB.x. DigitalAlarmPoint0	All MHI GPIBs are not working	Any valid digital alarm point on the highway	L1DIGALARM
DL.MGPIB.x. DigitalAlarmPoint1	Some MHI GPIB is not working	Any valid digital alarm point on the highway	L3DIGALARM
DL.MGPIB.x. TestMsgFile	Output file	Any valid file	stderr
DL.MGPIB.x. RestartProcedure	Filename of reboot process	None	$WDPF_HOME /dl/bin/dl_mhigpib -l 1
DL.MGPIB.x. LinkName	Name of the link	x = 1 or 2, depending on the subsystem number	MHI dls 1 gateway
DL.MGPIB.x. LocalHost	Drop where the MHIGPIB Interface data link will operate	Host name of WEStation configured as a Data Link Server with MHIGPIB Interface software	drop130
DL.MGPIB.x. DeviceName	Device driver name	Name of gpib port in /dev directory	/dev/dev1
DL.MGPIB.x. Timeout	Loop timeout in msec before next loop is started	100 to 4000 msec	500 msec

Table 4-1. MHIGPIB Interface Configuration File Parameters (Cont'd)

Parameter	Description	Valid Option	Default
DL.MGPIB.x. Delay	Loop delay time in msec. Tells DLS how many msec to wait after sending data before it starts to transmit data	0 to 32,767 msec	0
DL.MGPIB.x. IOTMO	I/O timeout - this is the maximum length of time the bus is allowed for transmission	0 = disabled; 1 = 10 usec; 2 = 30 usec; 3 = 100usec; 4 = 300 usec; 5 = 1msec; 6 = 3msec; 7 = 10 msec; 8 = 30 msec; 9 = 100 msec; 10 = 300 msec; 11 = 1 sec; 12 = 3 sec; 13 = 10 sec; 14 = 30 sec; 15 = 100 sec	9
DL.MGPIB.x. NEOI	No end of message identifier	0 = EOI disabled 1 = EOI enabled	1
DL.MGPIB.x. READDR	No unnecessary re-addressing of link	0 = no unnecessary addressing 1 = constant addressing	1
DL.MGPIB.x. EOS	End of string during read	0 = EOS disabled 1 = EOS enabled	0
DL.MGPIB.x. EOS_EOI	End of identity on end of string	0 = no EOI on EOS 1 = EOI on EOS	0
DL.MGPIB.x. EOS_SZ	End of string character size	0 = EOS 7 bits 1 = EOS 8 bits	0
DL.MGPIB.x. EOS_CHR	End of string character	Character = 0x00 - 0xFF	0
DL.MGPIB.x. BENDRD	Big or Little Endian (high/low order of bytes)	On Reads: 0 = Big Endian (high order of bytes) 1 = Little Endian (low order of bytes)	0
DL.MGPIB.x. BENDWR	Big or Little Endian (high/low order of bytes)	On Writes: 0 = Big Endian 1 = Little Endian	0

Table 4-1. MHIGPIB Interface Configuration File Parameters (Cont'd)

Parameter	Description	Valid Option	Default
DL.MGPIB.x. UNADDR	Unaddress (untalk, unlisten after every read/ write)	0 = no unnecessary UNT or UNL 1 = constant UNT or UNL	0
DL.MGPIB.x. BYTSPS	Sets speed of GPIB bus	1 (Normal) = 300 kbytes/sec	1
DL.MGPIB.x. Cable_len	Enables high-speed data transfer. Speed dependent on cable length in meters.	0 = Normal	0
DL.MGPIB.x. Xfer_Protocol	Enables DLS to switch between Normal and MHI protocol	0 = Normal (Unaddress after message) 1 = MHI (Unaddress before message)	0

4-3.2. Command Line Arguments

All of the CONFIG.DL parameters can be entered using command line arguments. At startup, command line arguments can be entered to overwrite the default values of any CONFIG.DL parameter.

Table 4-2 illustrates the command line arguments that can be used to overwrite the CONFIG.DL parameters in the CONFIG.DL file.

The command format is as follows:

```
dl_mhigpib [-l arg1] [-x arg2] [-t arg3]
```

Arguments in brackets, [], are optional.

Table 4-2. Command Line Arguments

Argument	Description	Function	Default
-l arg1	Arg1 is how the subsystem number gets set	Determines what configuration parameters are used in the CONFIG.DL file. For example, 1 = .1, 2 = .2, etc.	1
-x arg2	Configuration file	Specifies configuration file to use other than CONFIG.DL	CONFIG.DL
-t arg3	Test menu	Displays menu - choose test menu instead of program	None

4-3.3. MHIGPIB.GROUP File

The MHIGPIB.GROUP file is the group point file specified in the CONFIG.DL file for the **DL.MGPIB.x.PointGroupDefFile** parameter. Since this is a bidirectional link, there are two lists.

An example of a MHIGPIB.GROUP file is shown in Figure 4-2. Comment lines are indicated by a pound sign (#). Operation denotes the direction of data input/output as referenced to the DLS.

```
 # This is the input point group for drop 130.
 # Each group list contains the Point Name which is followed by the
 # G (Gain) and b (Bias), if the point is an analog point.
 # Hmax is maximum value of highway point.
 # Hmin is minimum value of highway point.
 # The formula for the input Gain is Nmax - Hmin/16384.
 # The formula for the input Bias is Hmin.

/GROUP:                name     = “MHI input data group”,
                       number   = 1,
                       operation= INPUT

# Point Name,          G,   b
 TESTAI01              1,   1
 TESTAI03              2,   2
 TESTAI04              3,   3
 TESTAI05              4,   4
 TESTAI06              5,   5
 TESTAI07              6,   6
 TESTAI08              7,   7
 TESTAI09              8,   8
 TESTAI10              9,   9
 TESTAI01              0,   0
 TESTDI02
 TESTDI03
 TESTDI04
 TESTDI05
 TESTDI06
 TESTDI07
 TESTDI08
 TESTDI09
 TESTDI10
 TESTDI01
```

Figure 4-2. Example of MHIGPIB.GROUP File

```
# This is the output point group for drop 130.
# Each group list contains the Point Name which is followed by the
# G (Gain) and b (Bias), if the point is an analog point.
# Hmax is maximum value of highway point.
# Hmin is minimum value of highway point.
# The formula for the output Gain is 16384/Hmax - Hmin.
# The formula for the output Bias is -16834 * Hmin/Hmax - Hmin.

/GROUP               name     = “MHI output data group”,
                     number   = 1,
                     operation= OUTPUT

# Point Name         G,   b
TESTAO02             1,   1
TESTAO03             2,   2
TESTAO04             3,   3
TESTAO05             4,   4
TESTAO06             5,   5
TESTAO07             6,   6
TESTAO08             7,   7
TESTAO09             8,   8
TESTAO10             9,   9
TESTAO01             0,   0
TESTDO02
TESTDO03
TESTDO04
TESTDO05
TESTDO06
TESTDO07
TESTDO08
TESTDO09
TESTDO10
TESTDO01
```

Figure 4-2. Example of MHIGPIB.GROUP File, Con’t.

TESTXXXX are the actual point names used to reference the highway. The entered sequence of point names in the file is the order of transmission, with all analog points being sent first, followed by the digital points. The first analog/digital point at the top of the list is MHI’s first analog/digital point sent to them in their block of data. The second analog/digital point from the top of the list is MHI’s second analog/digital point sent to them in their block of data. Similarly, the Nth, or last analog/digital point at the bottom of the list, is MHI’s Nth, or last, analog/digital point sent to them in their block of data.

Section 5. Communication Protocol

5-1. Section Overview

This section discusses the communication processes and message structure for the MHIGPIB Interface. It is divided into the following sections:

- Communication Processes (Section 5-2)
- MHI Messages (Section 5-3)
- Command Transfer Mode (Section 5-4)
- Data Transfer Mode (Section 5-5)

The MHIGPIB Interface uses elements of the IEEE 488.2 standard protocol. Refer to ANSI/IEEE 488.2-1987 for more information about IEEE 488.2 protocol.

5-2. Communication Processes

The GPIB bus is an eight-bit parallel half-duplex transmission command/data bus. This allows data to flow in both directions between the WDPF Westnet II Data Highway and the MHI DIASYS, but only in one direction at a time. The Westinghouse Data Link Server (DLS) controls the transmission of data. The DLS is the controller in charge (CIC). The CIC sets itself and the controlled units for each data transmission (cycle), and resets them after each cycle.

The "Handshake Principle" is a method of acknowledgment that data is either ready to be received or has already been transmitted or received. The CIC uses the Handshake Principle to reset the CIC and the controlled units after each data transmission. The Handshake Principle varies depending on whether the bus is in Command Transfer Mode (Section 5-4) or Data Transfer Mode (Section 5-5).

5-3. MHI Messages

The DLS and the MHI DIASYS communicate through the use of MHI messages. MHI messages can be IEEE 488 Uniline Control Signals (Section 5-3.1), IEEE 488 Multiline Command Messages (Section 5-4), or Data Packets (Section 5-5). Uniline Control Signals are used to determine whether the bus is in Command Transfer Mode or Data Transfer Mode.

5-3.1. Uniline Control Signals

In the MHIGPIB Interface, there are two categories of IEEE 488 Uniline Control Signals: Handshake Signals, and Interface Management (IM) Signals.

Handshake Signals

Handshake signals work together to guarantee that message bytes on the data lines are sent and received without transmission error. Each Uniline Handshake Signal corresponds to a physical control line on the bus. Three handshake lines asynchronously control the transfer of message bytes between the DLS and the MHI DIASYS. The three handshake signals are the following:

- NRFD (not ready for data) indicates that a device is ready or not ready to receive a message.
- NDAC (not data accepted) indicates that a device has or has not accepted a message byte.
- DAV (data valid) indicates that the signals on the data lines are stable and can be safely accepted.

Interface Management Signals

Interface Management (IM) signals manage the flow of information across the interface. The MHIGPIB Interface IM signals are the following:

- ATN (attention) - **True** indicates command transfer; **False** indicates data transfer.
- IFC (interface clear) - This signal is set **True** by the CIC to reset the interface and to become CIC of the bus. A True pulse is sent at startup and if multiple bus errors are detected.

- REN (remote enable) - This signal is set by the CIC and forces the controlled units into remote or local program mode. Under the MHI protocol handshake, REN is always True (commands can only come from the CIC).
- SRQ (service request) - This signal is set by the controlled unit(s). It is an interrupt signal to the CIC indicating that one or more controlled unit needs servicing. SRQ is not used under MHI protocol.
- EOI (end or identify) - This signal is set by the CIC and marks end of message string, or directs the controlled unit(s) to respond to a parallel poll to determine who is on line. Identify is not used under MHI protocol; however, the end signal is used.

Table 5-1 summarizes the Uniline control signals used by the MHIGPIB Interface.

Table 5-1. Uniline Control Signals

Signal	Description	Pin Number	Signal Category
EOI	End or Identify	5	Handshake End or IM
DAV	Data valid	6	Handshake
NRFD	Not ready for data	7	Handshake
NDAC	Not data accepted	8	Handshake
IFC	Interface clear	9	IM
SRQ	Service request	10	IM
ATN	Attention	11	IM
REN	Remote enable	17	IM

Each Uniline control signal corresponds to a physical control line on the bus. When the control signal is asserted, the line goes True.

The Uniline control signals use negative logic TTL voltage levels where:

True $\leq$ .08 V
False $\geq$ 2.0 V

In the normal free bus state, DAV is False, NRFD is False, and NDAC is True.

The mode of the IEEE 488 bus is determined by the Uniline ATN signal. If ATN is True, then the bus is in Command Transfer Mode. If ATN is False, then the bus is in Data Transfer Mode.

5-4. Command Transfer Mode

In the Command Transfer Mode, there is only one CIC, but there can be many controlled units. The CIC sends Multiline command messages to the controlled units, and the controlled unit(s) receive and process those messages.

In this mode, the MHIGPIB Interface uses four IEEE 488 Multiline command messages. They are as follows:

- UNL (unlisten) tells all controlled units and the CIC to stop listening to or receiving any further data messages
- UNT (untalk) tells all controlled units and the CIC to stop talking or sending any further data messages
- MLA (my listen address) tells the controlled unit or the CIC to listen if the address is LLLLL, where LLLLL can be a decimal value (0 - 30) or a hexadecimal value (0X0 - 0X1E).
- MTA (my talk address) tells the controlled unit or CIC to talk if the address is TTTTT once ATN goes low and NRFD is False, where TTTTT can be a decimal value (0 - 30) or a hexadecimal value (0X0 - 0X1E).

Table 5-2 summarizes the Multiline commands used by the MHIGPIB Interface.

Table 5-2. Multiline Commands

Command	Description	Data I/O (DIO) Lines 8 7 6 5 4 3 2 1	Category
UNL	Unlisten	0 0 1 1 1 1 1 1	Address data
UNT	Untalk	0 1 0 1 1 1 1 1	Address data
MLA	My listen address	0 0 1 L L L L L	Address data
MTA	My talk address	0 1 0 T T T T T	Address data
L = 0 - 30 (decimal); 0X0 - 0X1E (hexadecimal) T = 0 - 30 (decimal); 0X0 - 0X1E (hexadecimal) The address of the CIC is always 0.			

In the Command Transfer Mode, the CIC can set itself or any controlled unit to be either the Talker or the Listener. The CIC uses the Command Transfer Handshake Principle to reset the Talker and the Listener after each data transmission.

5-4.1. Command Transfer Handshake Principle

The Command Transfer Mode is initiated on the bus by the Uniline ATN signal being set True (logic level low voltage) and remaining so until all commands are sent. The Command Transfer Handshake procedure takes place as follows:

1. Bus Free State

   The CIC must wait for all controlled units to set the NRFD signal to False.

2. Multiline Command Applied to Bus

   The CIC sets the Multiline command on the bus.

3. Bus in Use (Command Ready)

   After the CIC determines that all Uniline control signals and Multiline commands are stable, it sets DAV to True. This notifies all controlled units that a command is ready to be picked up. The controlled units respond by setting NRFD to True.

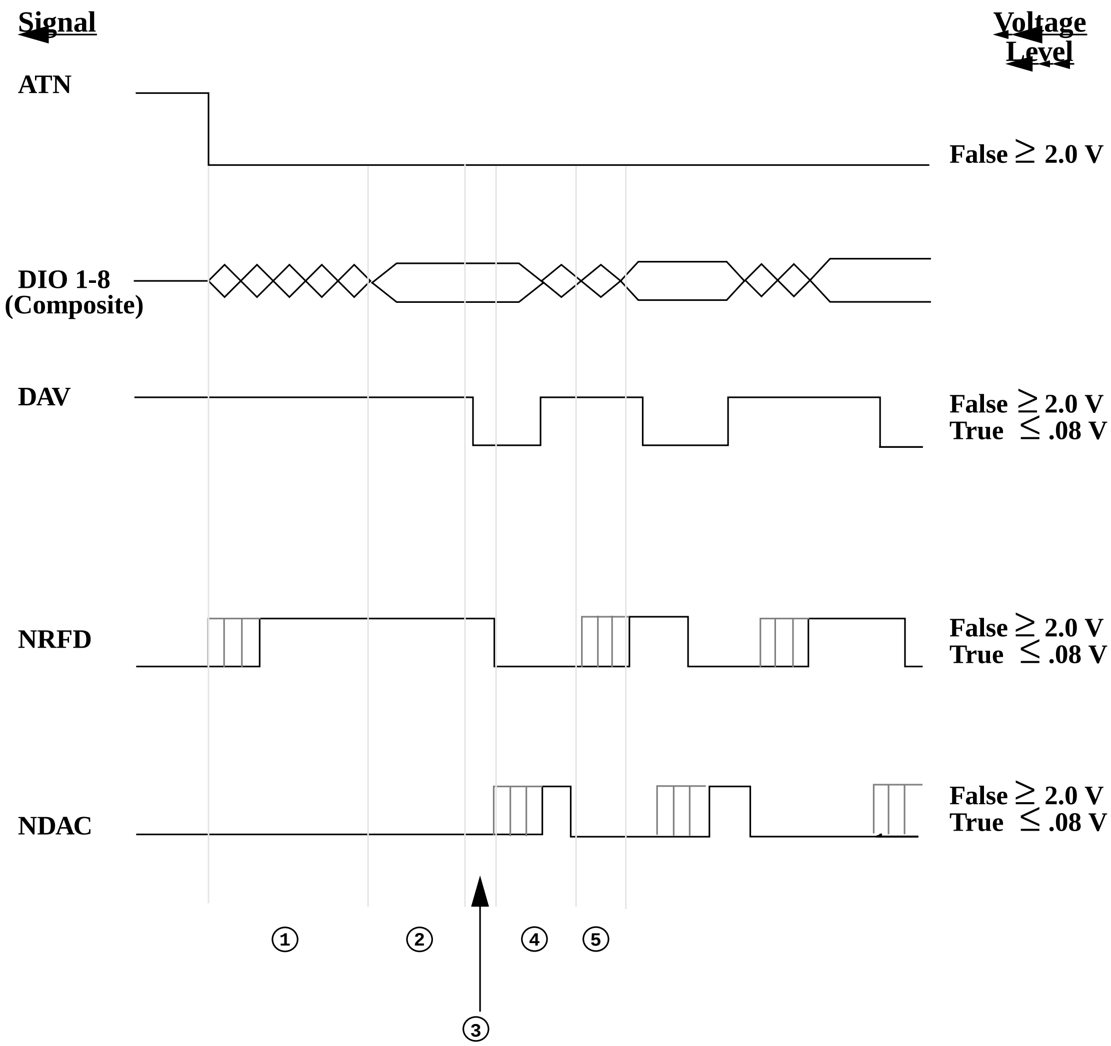
4. Command Accepted

   After each of the controlled units has received the Multiline command, they set the NDAC signal to False. This in turn causes the CIC to set the DAV signal to False.

5. Command Processed (Bus Free State)

   After the controlled units have processed the Multiline command, they first reset NDAC to True and then reset NRFD to False. This frees the bus when the controlled units are ready for the next command.

Figure 5-1 illustrates the Command Transfer Handshake Principle.



BUS ACTIVITY

① = Bus free

② = Command applied to bus

③ = Bus in use (command ready)

④ = Command accepted

⑤ = Command processed (bus free)

Figure 5-1. Command Transfer Handshake Principle

5-5. Data Transfer Mode

In the Data Transfer Mode, one IEEE 488 Multiline command is used to transfer data either from the DLS to the MHI DIASYS, or vice-versa. The command for transferring data is the DAB (Data Byte) command. When the DAB command is issued, data is transferred over the interface through data transmission packets.

There are two types of data transmission packets in the Data Transfer Mode, MHI DIASYS to DLS and DLS to MHI DIASYS.

5-5.1. MHI DIASYS to DLS Data Transmission Packet

The MHI DIASYS to DLS data packet is used when the DLS receives data from MHI (that is, when the DLS is the Listener). Figure 5-2 illustrates the MHI DIASYS to DLS data packet. It is arranged such that the bytes at the top of the figure are the bytes first sent out across the data link. The first three bytes of the message compose the message header and the last two bytes of the message compose the message tail.

Message Header

The message header is composed of the following three bytes:

- **Start Code** is a line check code; the only valid value is 0XAA
- **Sort Code** is an MHI code for distinguishing between send and receive packets; 2 = receive
- **Status Code** is an MHI code for determining packet validity; 1 = valid, and 0 = not valid

Message Tail

The message tail is composed of the following two bytes:

- **End Code** is a line check code; the only valid value is 0X55
- **Parity** is the longitudinal redundancy checksum (LRC) of all bytes between and including the Start Code and the End Code

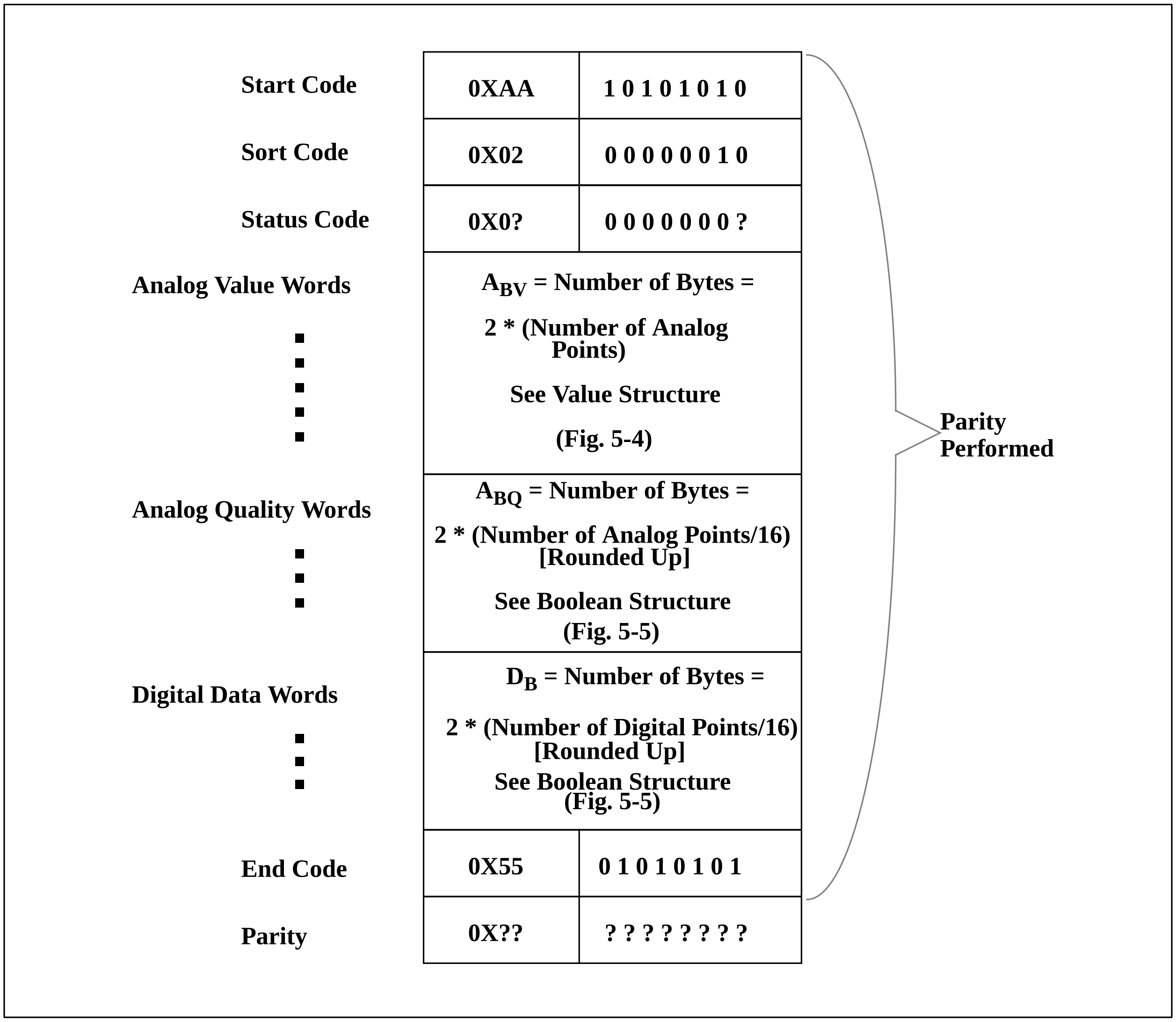


Figure 5-2. MHI DIASYS to DLS Data Packet

Structure elements of the MHI DIASYS to DLS data packet are illustrated in Figure 5-4 and Figure 5-5. Figure 5-4, Value Structure, details how analog values are sent. Figure 5-5, Boolean Structure, details how digital and analog quality values are sent.

Data Transfer Performance Parameters

For proper operation of the DLS, the number of analog bytes (A_{BV}), analog quality bytes (A_{BQ}), and digital bytes (D_B) must be less than the data transfer limit (DTL).

The following equation determines if the packet is too large for the configured I/O timeout. Refer to Table 4-1 for IOTMO configuration parameter values.

$$A_{BV} + A_{BQ} + D_B + 5 < DTL = (300{,}000 * IOTMO\ (sec))$$

Note

The DTL value is a theoretical maximum.
Actual performance may be less.

5-5.2. DLS to MHI DIASYS Data Transmission Packet

The DLS to MHI DIASYS data packet is used when the DLS transmits data to MHI (that is, when the DLS is Talker). Figure 5-3 illustrates the DLS to MHI DIASYS data packet. It is arranged such that the bytes at the top of the figure are the bytes first sent out across the data link. The first three bytes of the message compose the message header and the last two bytes of the message compose the message tail.

Message Header

The message header is composed of the following three bytes:

- **Start Code** is a line check code; the only valid value is 0XAA
- **Sort Code** is an MHI code for distinguishing between send and receive packets; 1 = send
- **Status Code** is an MHI code for determining packet validity; 1 = valid

Message Tail

The message tail is composed of the following two bytes:

- **End Code** is a line check code; the only valid value is 0X55
- **Parity** is the longitudinal redundancy checksum (LRC) of all bytes between and including the Start Code and the End Code

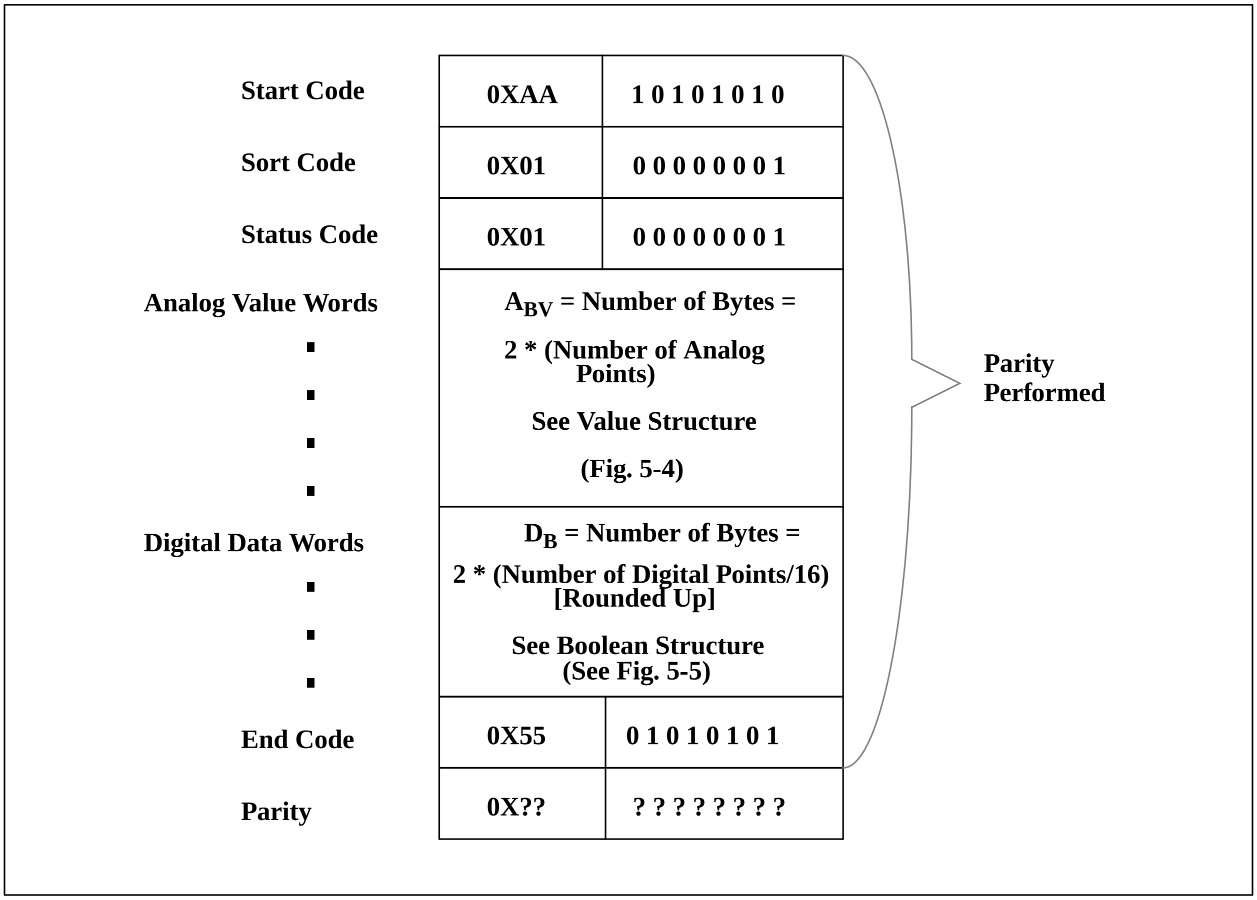


Figure 5-3. DLS to MHI DIASYS Data Packet

Structure elements of the DLS to MHI DIASYS data packet are illustrated in Figure 5-4 and Figure 5-5. Figure 5-4, Value Structure, details how analog values are sent. Figure 5-5, Boolean Structure, details how digital and analog quality values are sent.

Data Transfer Performance Parameters

For proper operation of the DLS, the number of analog bytes (A_{BV}), and digital bytes (D_B) must be less than the data transfer limit (DTL).

The following equation determines if the packet is too large for the configured I/O timeout. Refer to Table 4-1 for IOTMO configuration parameter values.

$$A_{BV} + D_B + 5 < DTL = (300{,}000 * IOTMO\ (sec))$$

Note

The DTL value is a theoretical maximum.
Actual performance may be less.

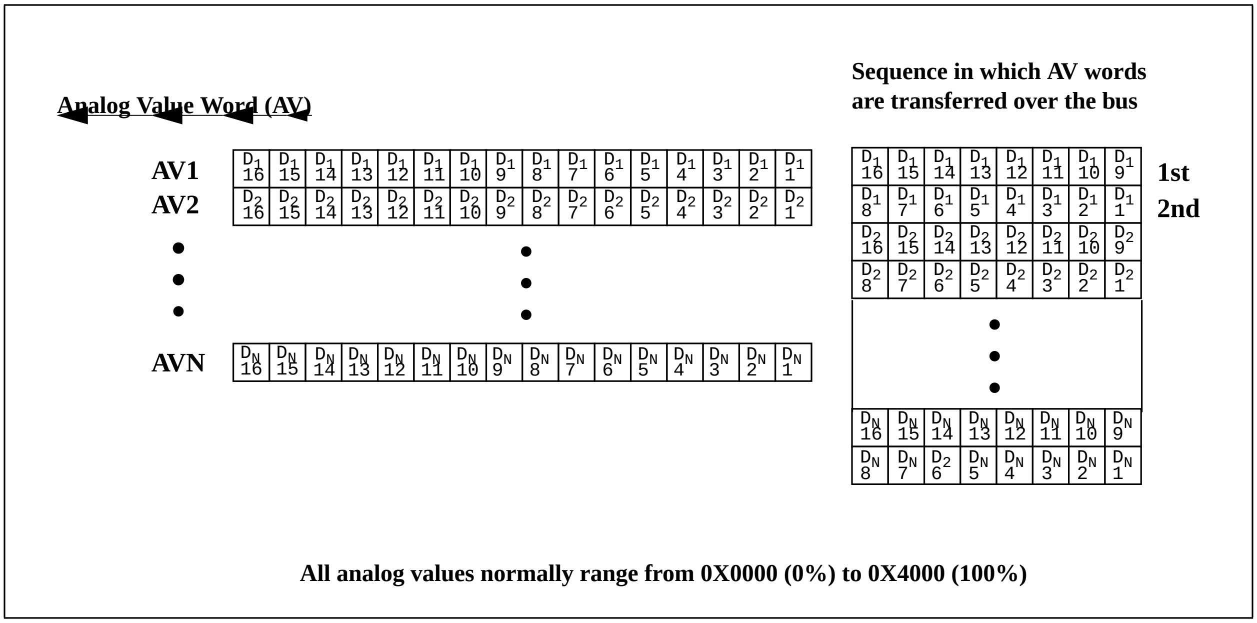


Figure 5-4. Value Structure

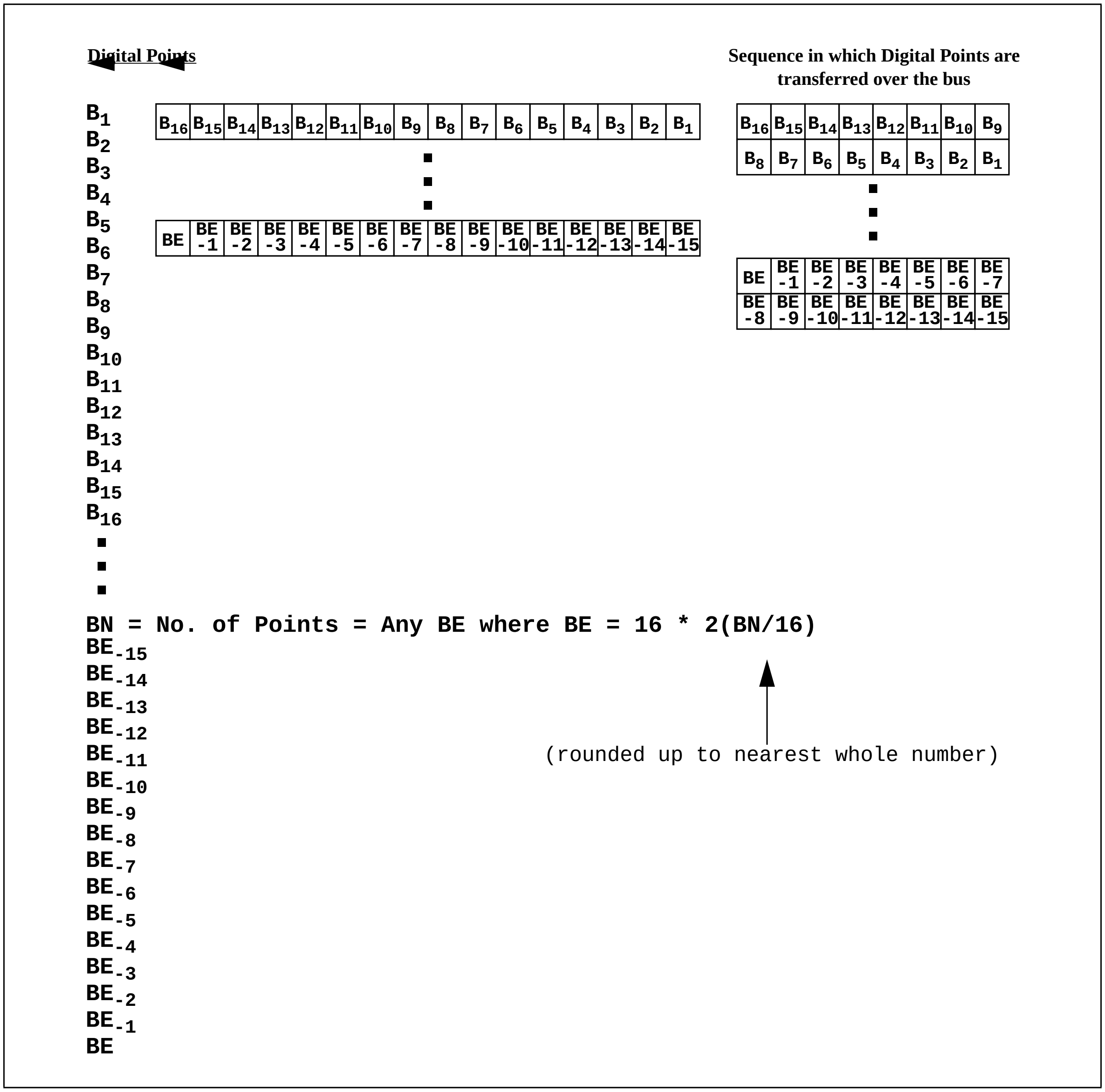


Figure 5-5. Boolean Structure

5-5.3. Data Transfer Handshake Principle

Data Transfer Mode is initiated on the bus by the Uniline ATN signal being set False (logic level high voltage) and remaining so until all data is sent. The Data Transfer Handshake Procedure takes place as follows:

1. Bus Free State

   The Talker waits for all Listeners to set NRFD to False.

2. Data Applied to Bus

   The Talker sets its data byte on the bus.

3. Bus in Use (data ready)

   After the Talker verifies that all the control signals and data bytes are stable, it sets DAV True which notifies Listeners to pick up the data. This causes all Listeners to set NRFD True.

4. Data Accepted

   After the Listeners receive the data byte, they set NDAC to False. This in turn causes the Talker to set DAV False.

5. Data Processed (bus ready)

   After the Listeners have processed the data, they first reset NDAC to True and then reset NRFD to False. This frees the bus when the controlled units are ready for the next command.

The process ends in the MHI Handshake Mode when EOI is set true on the last byte. A command is the next expected event on the bus.

Figure 5-6 illustrates the Data Transfer Handshake Principle.

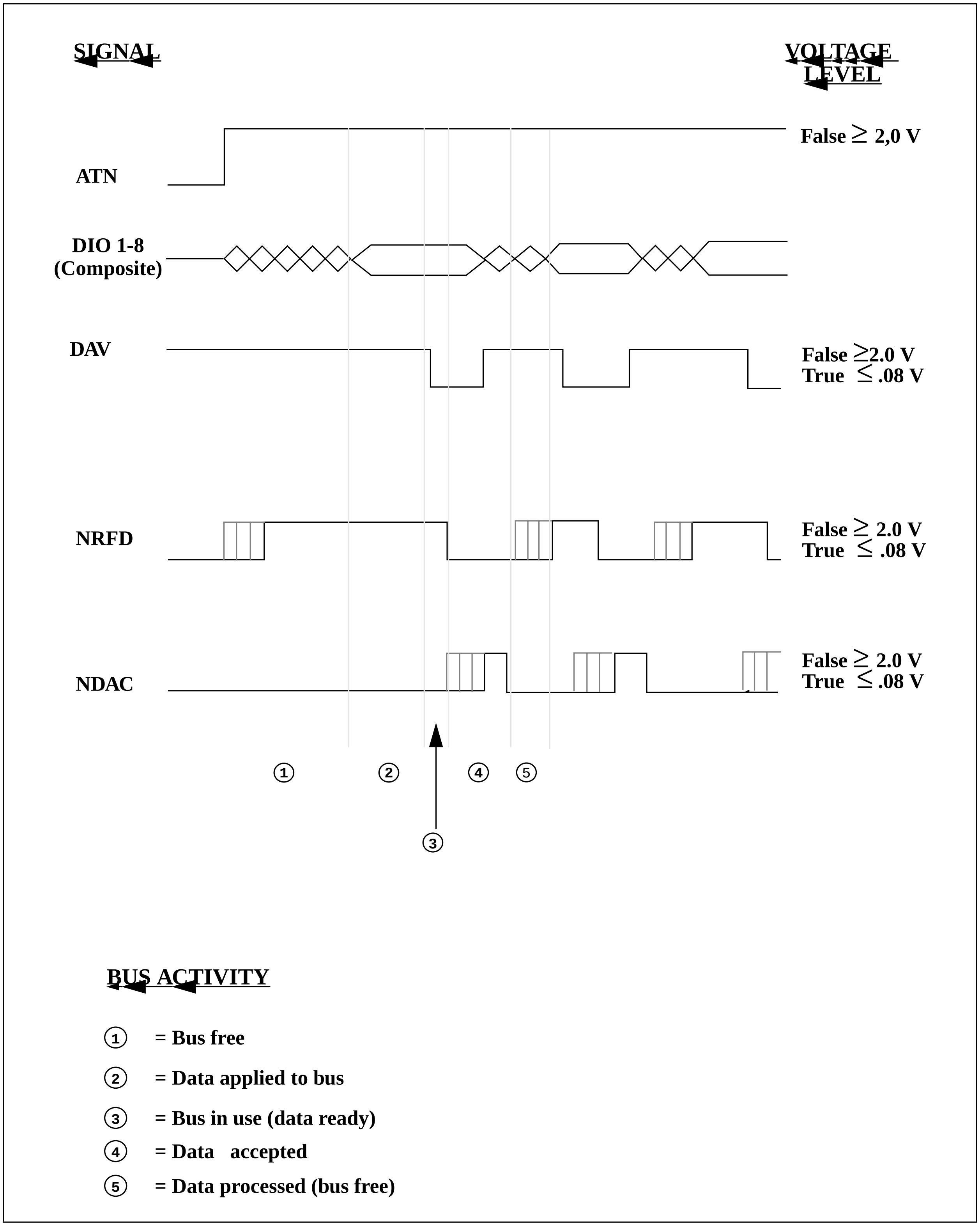


Figure 5-6. Data Transfer Handshake Principle

Section 6. Interface Operation

6-1. Section Overview

This section discusses the MHIGPIB Interface status. It is divided into the following sections:

- Startup (Section 6-2)
- Interface Operations (Section 6-3)
- Shutdown (Section 6-4)

6-2. Startup

The MHIGPIB Interface process is started when the WEStation is booted. It can also be restarted by using the Query/Download function at an Operator WEStation. Refer to “Engineering WEStation User’s Guide” (U0-8200) for more information on the Query/Download function.

The **startup.dl** file in the $WDPF_HOME/dl/init directory is configured to start the MHIGPIB Interface.

The $WDPF_HOME and $WDPF_PDIR environment variables should be set before starting the software. (These are mentioned here for verification purposes only. No new environment variables need to be set for this software.)

6-2.1. Initialization

The MHIGPIB Interface software performs the following initialization tasks:

- Reads the CONFIG.DL configuration file
- Opens the Sun™ Highway Controller (SHC) memory to access process point data.
- Initializes the error reporting mechanism.
- Reads the MHIGPIB.GROUP file.
- Verifies the validity of all the point groups.

6-2.2. Process Point Quality upon Startup

Startup of the MHIGPIB Interface software temporarily changes the quality of the process points to which the software is writing to Bad quality. As the software progresses through the commands and starts mapping data to process points, the quality of these points is restored to Good quality. This process provides confirmation that the interface software is functioning correctly and is receiving valid messages.

6-3. Interface Operations

Diagnostic capabilities of the MHIGPIB Interface software include the following:

- Functions of the 32-bit link status point
- Configuration parameters that enable error and status messages to be sent to a screen or to a specified output file
- Drop fault report that identifies an error

6-3.1. Link Status Point

The status of the MHIGPIB Interface is determined by the value of a packed digital (PB) point which is broadcast on the WDPF highway. This point is an originated point on the WEStation and must be specified in the CONFIG.DL configuration file in the **DL.MGPIB.x.LinkStatusPoint** parameter. If no point is specified in the CONFIG.DL file, then this feature is disabled, an informational message is logged during interface startup, and no status is reported.

The link status point has a 32-bit value. The current message count, last reported error code, and subsystem status are mapped into this point. Description of each applicable bit is provided below.

- Bits 0 - 7 indicate the Link Status.

Binary Status	**Decimal Status**	**Description**
0 0 0 0 0 0 0 0 =	0 =	Shutdown
0 0 0 0 0 0 0 1 =	1 =	Alive
0 0 0 0 0 0 1 0 =	2 =	Enabled
0 0 0 0 0 0 1 1 =	3 =	Connected
0 0 0 0 1 0 0 0 =	8 =	Backup

- Bits 8 - 15 indicate the Error Codes.

Binary Code	**Decimal Code**	**Description**
0 1 1 0 1 1 0 1 =	109 =	SHC initialization error
1 1 0 0 1 0 0 1 =	201 =	Device initialization error
1 1 0 1 0 1 0 0 =	212 =	Received invalid message
1 1 0 1 0 1 0 1 =	213 =	Invalid message length
1 1 0 1 0 1 1 1 =	215 =	Error in reading the message
1 1 0 1 1 0 0 0 =	216 =	Message timed out
1 1 0 1 1 0 0 1 =	217 =	Error in waiting for the message
1 1 0 1 1 0 1 1 =	219 =	Error in reading from the SHC
1 1 0 1 1 1 0 0 =	220 =	Error in writing to the SHC
1 1 0 1 1 1 1 0 =	222 =	Message checksum error
1 1 0 1 1 1 1 1 =	223 =	Invalid message type

- Bits 16 - 19 are not applicable.
- Bits 20 - 32 are used to indicate the message count.

6-3.2. Error Reporting

The software indicates errors by displaying status messages to the console or to a specified output file. The software uses the Syslog Error Logging Utility under UNIX®.

Error and status messages are sent to the **syslog** and are reflected on the General Message Display of the Operator Station (since the DLS drop normally does not have a monitor).

The messages displayed depend upon the warning level (priority code) specified by the user in the CONFIG.DL file. If the priority code is set to 3, then all messages below and including level 3 are displayed. For this interface, the highest level of error reporting available is 3. Level 6 would display informational messages regarding point group sizes and various processes started, as well as all messages of levels 0 through 5.

A configuration parameter is also used to enable the writing of error and status messages to a specified output file. This feature is used for debugging and is not normally enabled. The various warning levels are indicated in Table 6-1. The default logging level is 3.

Table 6-1. Priority Levels for Error Logging

Message	Priority Codes	Condition
LOG_DEBUG	7	Debug level messages
LOG_INFO	6	Informational messages
LOG_NOTICE	5	Normal but significant conditions
LOG_WARNING	4	Warning conditions
LOG_ERR	3 (Default)	Error conditions
LOG_CRIT	2	Critical conditions
LOG_ALERT	1	Action must be taken immediately
LOG_EMERG	0	System is unusable

6-3.3. Drop Fault

Although the software is designed to operate indefinitely, it may encounter a problem from which it cannot recover. If the subsystem detects an unrecoverable error, then it declares a drop fault and aborts.

Drop faults are reported for the following conditions:

- Error parsing the CONFIG.DL configuration file.
- Error in accessing the SHC or System Point Directory (SPD)
- Error creating control system parameters shared memory/semaphore
- Error in subsystem already running

A drop fault does not affect the other subsystems on the drop. However, a drop fault does affect the quality of process points to which the interface software is writing. These will be invalid (bad quality).

The drop fault description is shown on the System Status Diagram display (refer to “Self-Test Diagnostics” (M0-0003) for further information regarding the System Status Diagram).

The drop fault report consists of the current drop fault code, fault ID, and fault parameters 1 and 2. The fault ID identifies the subsystem that reported the fault (MHIGPIB Interface), and the fault parameter identifies the reason for failure (for example, SHC error, missing configuration file, or communication device failure).

Fault parameter 1 identifies the device that failed. Fault parameter 2 is the error code of the device that failed (refer to “Self-Test Diagnostics” (M0-0003) for further information regarding fault codes.

The possible fault codes are listed in Table 6-2. The fault code for Data Links is **190**. The fault ID for the MHIGPIB Interface is **25**.

Table 6-2. Drop Fault Parameters

Fault Parameter 1	Fault Parameter 2	Description
3	errno.h	System call error
6	shc.err.h	SHC error
14	0	Unscheduled exit

6-4. Shutdown

The system can be shut down using the Data Link Server Manager. Refer to "WEStation Data Link Server User's Guide" (U0-8700) for more information.

Upon shutdown, the quality of the process points to which the MHIGPIB Interface software is writing changes temporarily to Bad quality.

The MHIGPIB Interface software is designed to operate indefinitely. If it fails for any reason, then an error message is displayed, indicating the reason for failure (see Section 6-3.2). If it encounters an unrecoverable error, then a drop fault is reported (see Section 6-3.3).

Appendix A. Installing the NI-488.2M Driver

A-1. Installation Procedure

The NI-488.2M driver (Version 1.3b or later) requires a Solaris 2.x operating system. For this version of the MHIGPIB Interface, use the following installation procedure to install the software driver. User entries appear in **`bold`** typeface, and screen responses appear in regular typeface. For additional information or future revisions, follow National Instrument Corporation's installation material.

Prepare for Installation

1. Do a remote login from the main console unit to the data link drop, using the same password that was used to log into the main console.

   ```
   rlogin sunxx
   password: <login password>
   ```

2. Log on as super-user (root), using the system password for the DLS.

   ```
   su root
   password: <system password>
   ```

3. Access the /tmp directory by entering the following command:.

   ```
   cd /tmp <Return>
   ```

4. Remove any version of NI-488.2M prior to Version 1.3 from the DLS disk drive by entering the following command:.

   ```
   pkgrm <Return>
   ```

The following screen appears:

```
The following packages are available:
1   NICsbgpib    NI-488.2M SB-GPIB Driver for Solaris 2.x
      (sparc) 1.3
2   SUNWadmap    System & Network Administration Application
     (sparc) 6.0.9
3   SUNWadmfw     System & Network Administration Framework
      (sparc) 6.0.8
4   SUNadmr       System & Network Administration Root
      (sparc) 6.0, REV = 2.0
5   SUNWaudio    Audio applications
      (sparc) 3.0, REV = 1.1.1
6   SUNWbcp      Binary Compatibility
      (sparc) 11.5.0, REV = 2.0.18
7   SUNWbtool    CCS tools bundled with SunOS
      (sparc) 11.5.0, REV = 2.0.18
8   SUNWcar      Core Architecture, (Root)
      (sparc.sun4c) 11.5.0, REV = 2.0.18
9   SUNWcar.2    Core Architecture, (Root)
      (sparc.sun4c) 11.5.0, REV = 2.0.18 PATCH=37
10  SUNWcg6      GX (cg6) Device Driver
      (sparc.sun4c) 11.5.0, REV = 2.0.18
... 52 more menu choices to follow;
<RETURN> for more choices, <CTRL-D> to stop display.
```

5. Since the NI-488.2 package (NICsbgpib) is visible, enter **<CTRL-D>**. The following interactive prompts appear (user response is indicated in ‘bold’ print):

   - `Select the package(s) you wish to process (or ‘all’ to process all packages). The options are [?,??,q]; the default is ‘all.’`

     Enter: **1**

- `The following package is currently installed:`
  `NICsbgpib  NI-488.2M SB-GPIB Driver for Solaris 2.x (sparc) 1.3`

  `Do you wish to remove this package?  The options are [y,n,?,q].`

  Enter: **`y`**

- `## Removing installed package instance <NICsbgpib>`

  `This package contains scripts which will be executed with superpermission during the process of removing this package.`

  `Do you want to continue with the removal of this package?  The options are [y,n,?,q].`

  Enter: **`y`**

When the removal process is complete, the following message will appear on the screen:

```
Removal of <NICsbgpib> was successful.
```

Install the Software Driver

6. Insert the NI-488.2M Distribution Disk for SB-GPIB and Sun SPARCstation, Version 1.3b or later. Enter the following command: **`volcheck`**

7. Cancel the disk window just opened by entering the following command:

   `cancel <Return>`

8. Add NI-488.2M, Version 1.3b or later, from the DLS disk drive. Enter the following command:

   `pkgadd -d /vol/dev/diskette0/unlabeled  <Return>`

9. The following interactive prompts appear:

- ```
  The following packages are available:
  1  NICsbgpib  NI-488.2M SB-GPIB Driver for Solaris
  2.x (sparc) 1.3b.

  Select packages you wish to process (or ‘all’ to
  process all packages). The options are : [?,??.q};
  the default is ‘all.’
  ```

  Enter: **1**

- ```
  Processing package instance <NICsbgpib> from </vol/
  dev/diskette0/unlabeled>

  NI-488.2M SB-GPIB Driver for Solaris 2.x (sparc)
  1.3b

  Copyright (c) 1993
  National Instruments Corp.
  All rights reserved.

  The selected base directory </opt/NICgpib> must
  exist before installation is attempted.

  Do you want this directory created now?  The options
  are: [y,n,?,q}.
  ```

  Enter: **y**

- ```
  Using </opt/NICgpib> as the package base directory.

  NI-488.2M SB-GPIB Driver for Solaris 2.x

  This installer will automatically remove old
  versions of the driver, ib, from /usr/kernel/drv
  and /kernel/drv.  Also, if module ib is already
  bound, this installer will execute rem_drv ib.

  Do you want to continue with the installation?  The
  options are: [y,n,?]; the default is ‘y.’
  ```

  Enter: **y**
  ```

- `Where should the include file ugpib.h be installed? The options are: [?,q]; the default is /usr/include/sys.`

  Enter: **/tmp**

- `Where should the utilities ibconf, ibic, and ibtsta be installed? The options are: [?,q]; the default is /usr/bin.`

  Enter: **/usr/bin**

- `Where should libgpib.a and libgpib.so be installed? The options are: [?,q]; the default is /usr/lib/.`

  Enter: **/tmp**

```
## Processing package information.
## Processing system information.
  2 package pathnames are already properly
installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already
installed.
```

- `The following files are already installed on the system and are being used by another package: /tmp <attribute change only>`

  `Do you want to install these conflicting files? The options are: [y,n,?,q].`

  Enter: **y**

  ```
  ## Checking for setuid/setgid programs.
  ```

- `This package contains scripts which will be executed with superuser permission during the process of installing this package.`

  `Do you want to continue with the installation of this package? The options are: [y,n,?,q].`

  Enter: **y**

- `Installing NI-488.2M SB-GPIB Driver for Solaris 2.x as <NICsbgpib>.`

When the installation process is complete, the following message will appear:

```
Installation of <NICsbgpib> was successful.
```

10. Remove floppy disk by entering the following command:

`eject <Return>`

Test Installation of Driver

11. Begin installation test by entering the following command:

ibsta `<Return>`

The following information appears on the screen:

```
National Instruments GPIB Software Installation Test,
Part A
Copyright (c) 1994 National Instruments Corp.  Version
1.0
All rights reserved...
```

The installation test performs the following functions:

- Checks the installation and operation of the GPIB software.
- Requires no interaction with the user.
- Assumes there is a GPIB board in the system, with special file /dev/gpib0.
- Takes approximately 10 minutes to complete.

The following message appears on the screen:

```
DISCONNECT ALL GPIB CABLES FROM THE GPIB BOARD.
```

12. Press **<Return>** to begin tests.

    The following message appears:

    ```
    Test in progress...
    ```

    When the test is complete, the following message appears:

    ```
    The software is correctly installed.
    ```

    The NI488.2M software driver installation is complete.